z/VM Concepts and Introduction

Sam Cohen
Levi, Ray & Shoup, Inc.

Sam.Cohen@lrs.com

June 20, 2024

This presentation is designed to present an overview of z/VM to z/OS and *nix-oriented systems programmers. The presenter is assumed to be well versed in z/VM concepts and terminology and should be able to understand terms and concepts of z/OS and Linux.

Agenda

Operating System Design
z/VM Concepts
z/VM Components
CMS Filesystems
Common CP/CMS Commands
More Detailed Topics
Guest Operating System Considerations under z/VM

Operating System Design

Operating System Design

- Every computer operating system has two fundamental design principles
- Number of executing tasks (1, >1)
- Work Dispatch mechanism (Priority/Interrupt, Timeslice)

Operating System Design

- Every computer operating system has two fundamental design principles
- Number of executing tasks (1, >1)
- Work Dispatch mechanism (Priority/Interrupt, Timeslice)

Design\Operating System	PC-DOS	Windows	*nix	z/OS, VSE^n	z/VM
Number of executing tasks	1	Multiple	Multiple	Multiple	Multiple
Dispatch mechanism	Priority	Timeslice	Timeslice	Priority	Timeslice

z/VM Concept

z/VM Concept

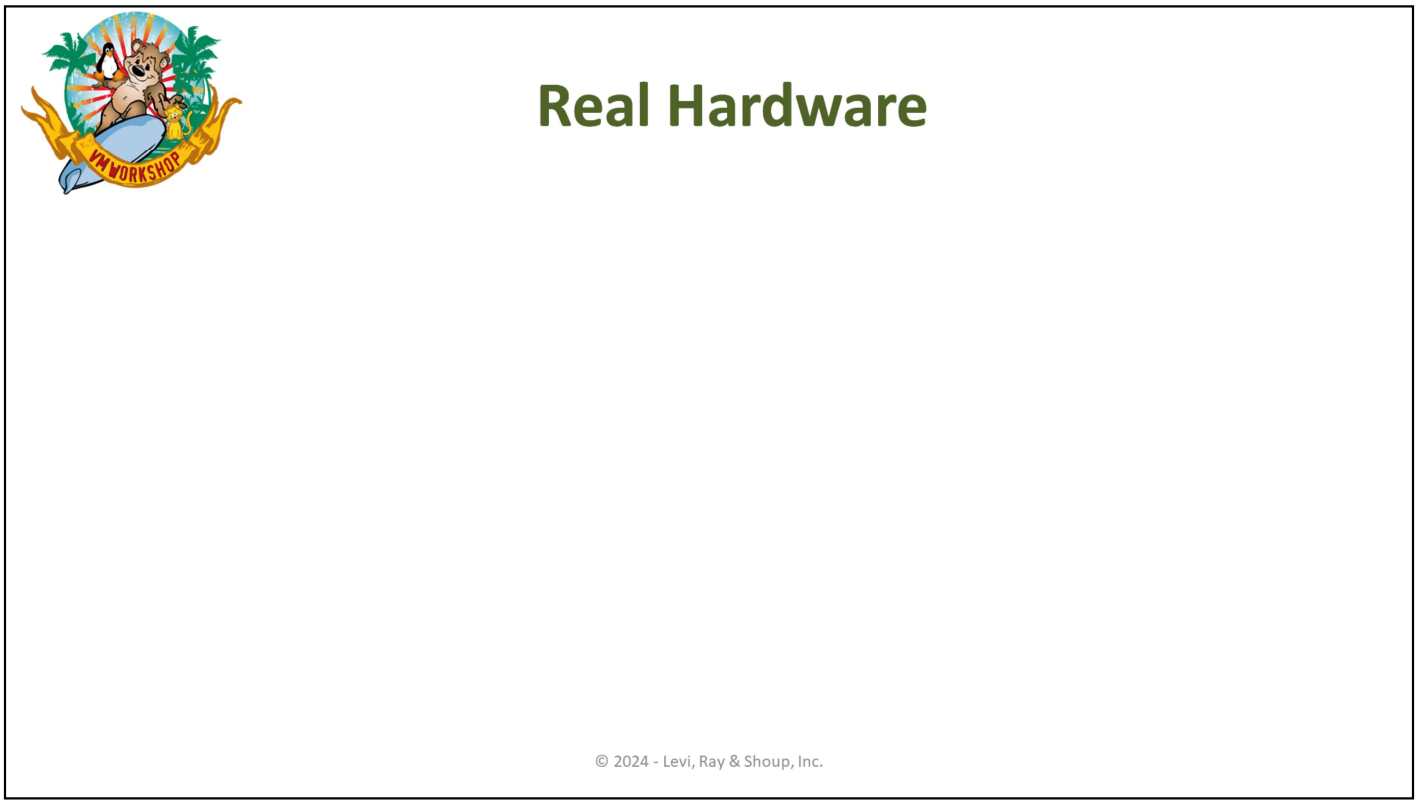
Back in the 1960s, when computers were physically large and expensive, IBM needed a way to virtualize a computer so multiple releases of its operating systems (OS (now z/OS) and DOS (now VSE^{n})) could be developed and tested. CP-67 (now z/VM) was developed to support that concept.

z/VM Concept

- z/VM Provides for the Definition and Management of Virtual Computers
  - Management of Real Hardware in support of Virtual Machines ("users" in z/VM terminology)
  - **A Virtual Machine has only virtual hardware**
  - Definition of Virtualized Hardware
    - Dedicated Real Hardware
    - Allocation of Shared Real Hardware
    - Dedicated Virtual Hardware
    - Allocation of Shared Virtual Hardware

Although a Virtual Machine uses real computer resources, it has no idea what the "real" world looks like, only what it can see. Everything in a virtual machine looks real to the guest operating system.

A Virtual Machine uses real hardware resources, but even with dedicated devices (like a tape drive), the virtual address of the tape drive may or may not be the same as the real address of the tape drive. Hence, a virtual machine only knows virtual hardware that may or may not exist in the real world.

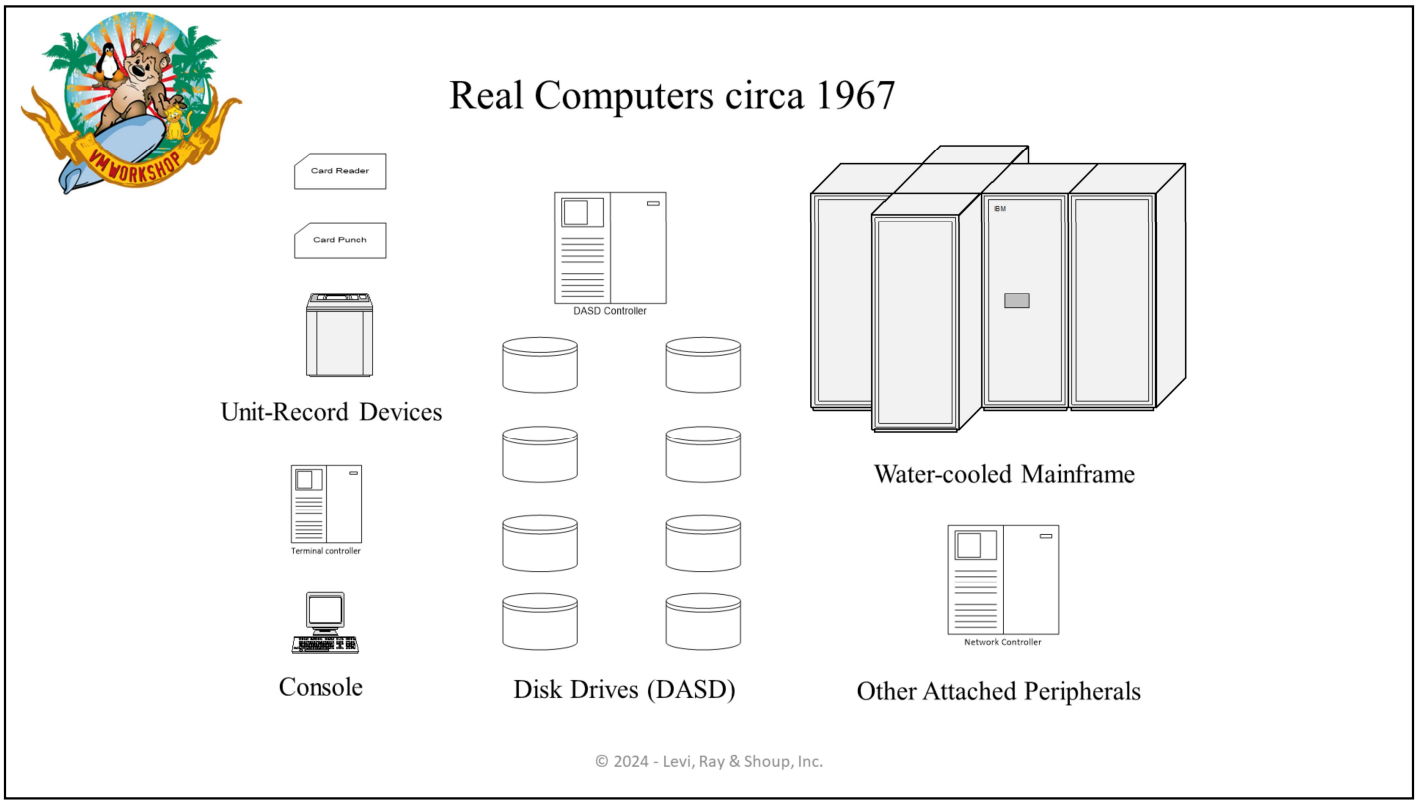


z/VM has its roots in a laboratory project that “escaped” and was known as CP67. Its paradigm harkens to computers and concepts from the S/360 and S/370 days.

Real Computers circa 1967

So, what made up a computer in 1967? A computer had:

1) A Water-Cooled mainframe computer, with a certain amount of memory (also known as Central Storage)
2) Unit-Record Devices…card reader, card punch and line printer
3) A console for interfacing with the computer
4) Drum or disk drives (known as Direct Access Storage Devices or DASD)
5) Other devices attached via channel cables



So, what made up a computer in 1967? A computer had:

1) A Water-Cooled mainframe computer, with a certain amount of memory (also known as Central Storage)
2) Unit-Record Devices…card reader, card punch and line printer
3) A console for interfacing with the computer
4) Drum or disk drives (known as Direct Access Storage Devices or DASD)
5) Other devices attached via channel cables

Virtualized Hardware via CP

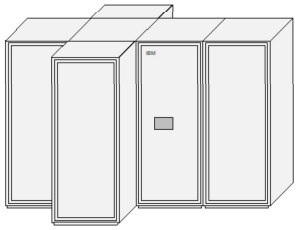
CP (Control Program)

- Is one of z/VM's components
- Creates virtual computing resources for a virtual machine
- Manages real hardware in support of those virtual computing resources

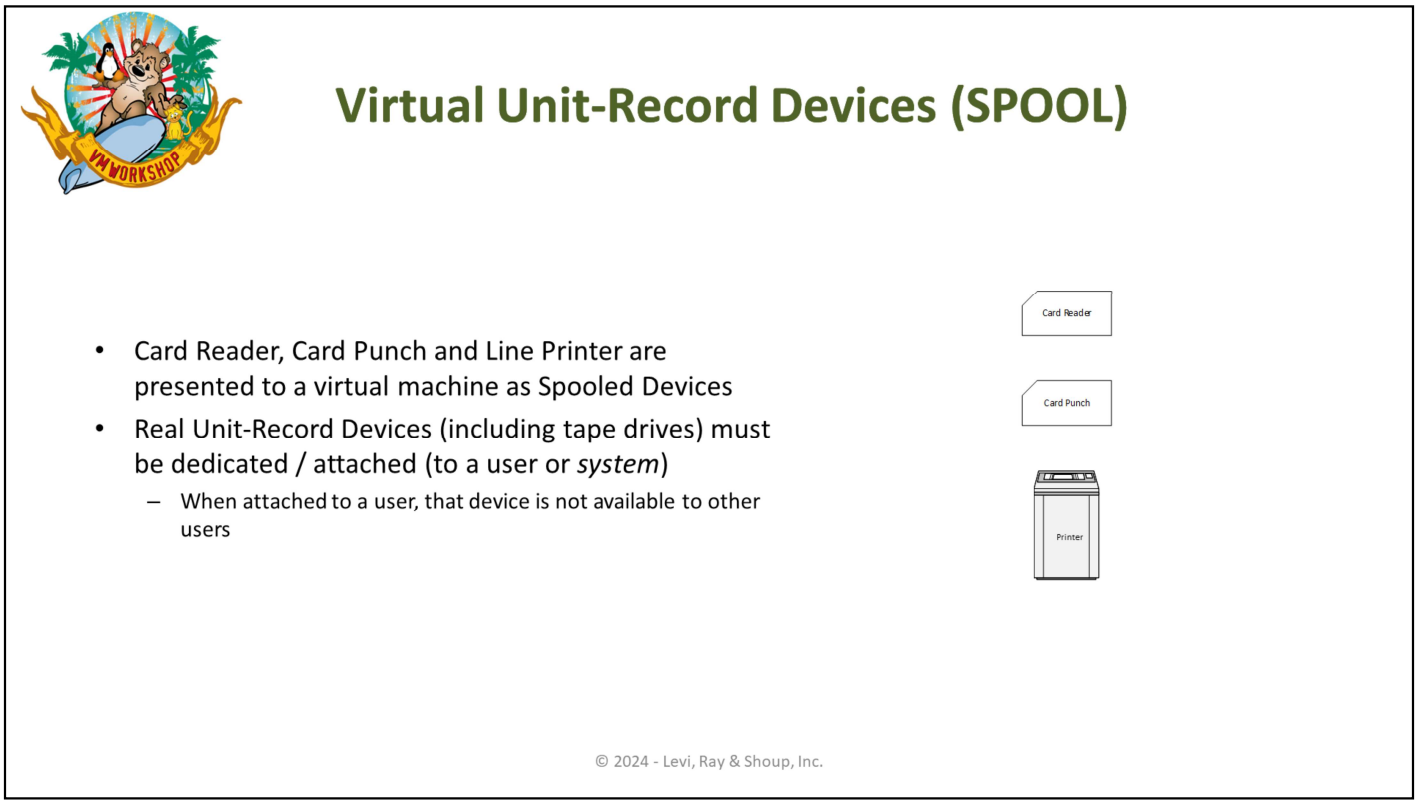
CP is the main focus in this presentation. Its job is to manage hardware (real or virtual) and virtual machines.

Virtual Processors (CPU)

- CP can dedicate real processors or share processors between virtual machines
- CP can present more virtual processors than are available either physically or logically
- Virtual processors are not the same as Logical processors
  - Physical processors – the number of actual problem processors on the hardware
  - Logical processors – the number of problem processors assigned to a Logical Partition (LPAR)
    - Cannot exceed the number of physical CPUs
  - Virtual processors – the number of problem processors assigned to a Virtual Machine



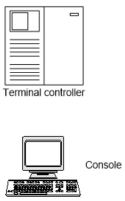
This series of charts show how CP takes its knowledge of real hardware and allows creation of virtual hardware



Did you know that SPOOL is an acronym? Simultaneous Peripheral Operations On-Line

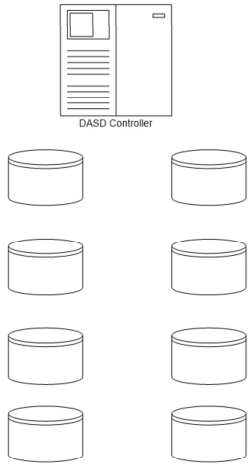
Console Device

- Virtual Console does not have to be physically connected to a real terminal
- If not connected to a real terminal, the virtual machine is considered to be “disconnected”



Disk Devices

- Disk space can be contiguous subsets of physical disk (termed "minidisk"; similar to disk partitions in the x86 world)
- Disk space can also be "virtual storage" – carved out of memory (similar to a PC-DOS RAM Disk)
- Disk space can be on Count-Key-Data (CKD) or Open Systems (SCSI) devices
  - CKD devices have variable-block physical allocation
  - SCSI devices have fixed-block physical allocation (512 bytes/block)
- Minidisks can be shared between multiple virtual machines
- A minidisk cannot span physical disks



Other Physical and Virtual Devices

- Examples of other Physical Peripherals:
  - Physical connections between LPARs/systems
  - FICON Directors
  - SAN Switches
- Virtual Peripherals
  - CTC (3088/ESCON/FICON), IUCV
  - 3270 and ASCII Terminals (GRAFs)
  - Guest LANs (Hipersockets, QDIO Ethernet)
  - Coupling Facilities and Coupling Links
  - Network Switches (Layer 2 or Layer 3 with VLAN awareness, LAP-C)

A Programmer's Viewpoint

Taking the mystery out of it, a Virtual Machine is implemented as

- To a z/OS sysprog, an address space
- To a Linux sysprog, a process

- But with a high degree of isolation from any other address space/process

- Each virtual machine is isolated from every other virtual machine, unless resource sharing is permitted by system definitions.

System Definitions

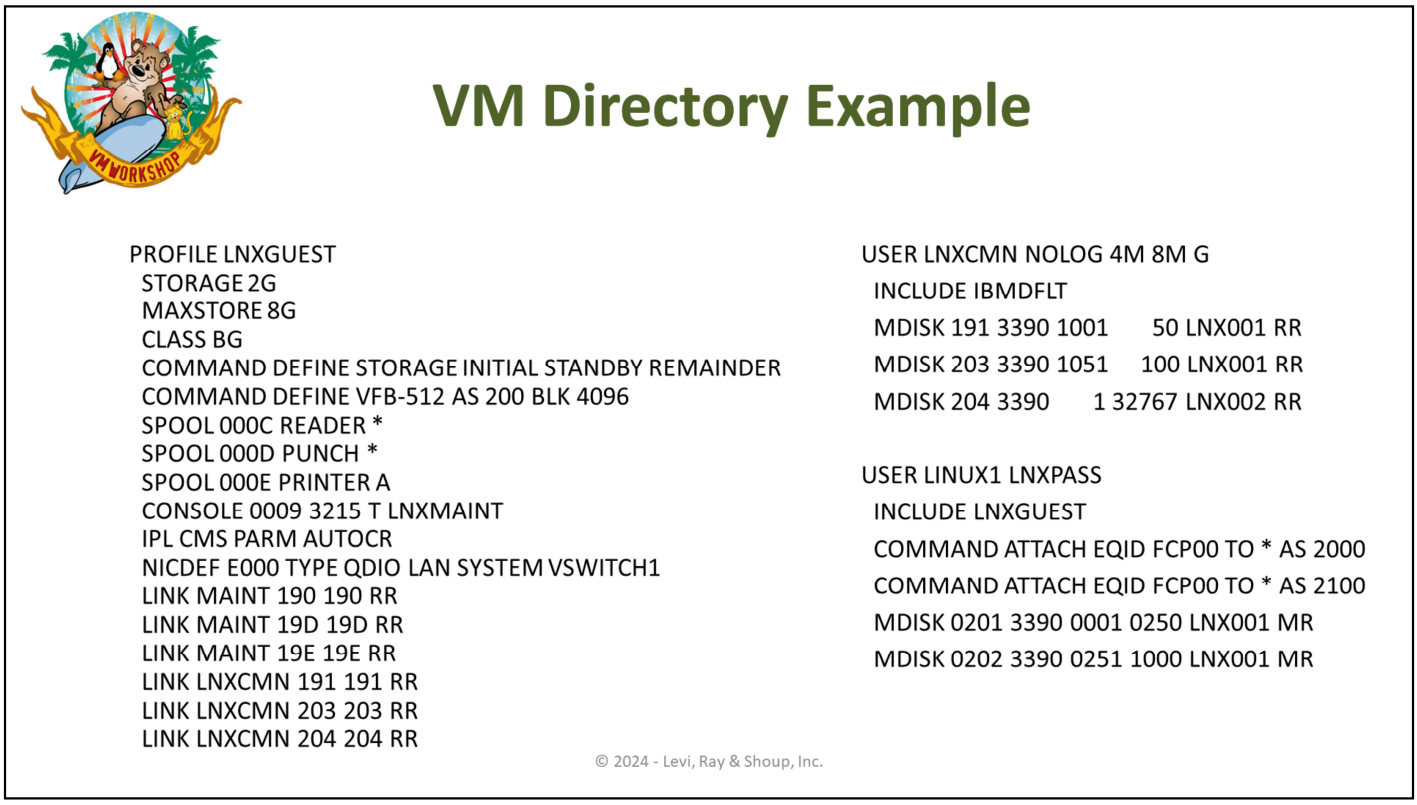
System Configuration File

- At system startup (IPL), attached devices are scanned and identified automatically
- If a device cannot identify itself or needs additional information (such as printer characteristics), then the device is defined in the System Configuration file
- This file also contains additional data, including
  - Determination of System Name
  - Definitions of virtual hardware
  - Live Guest Relocation information
  - Redefinition of CP Command Classes
  - Enablement of optional system features

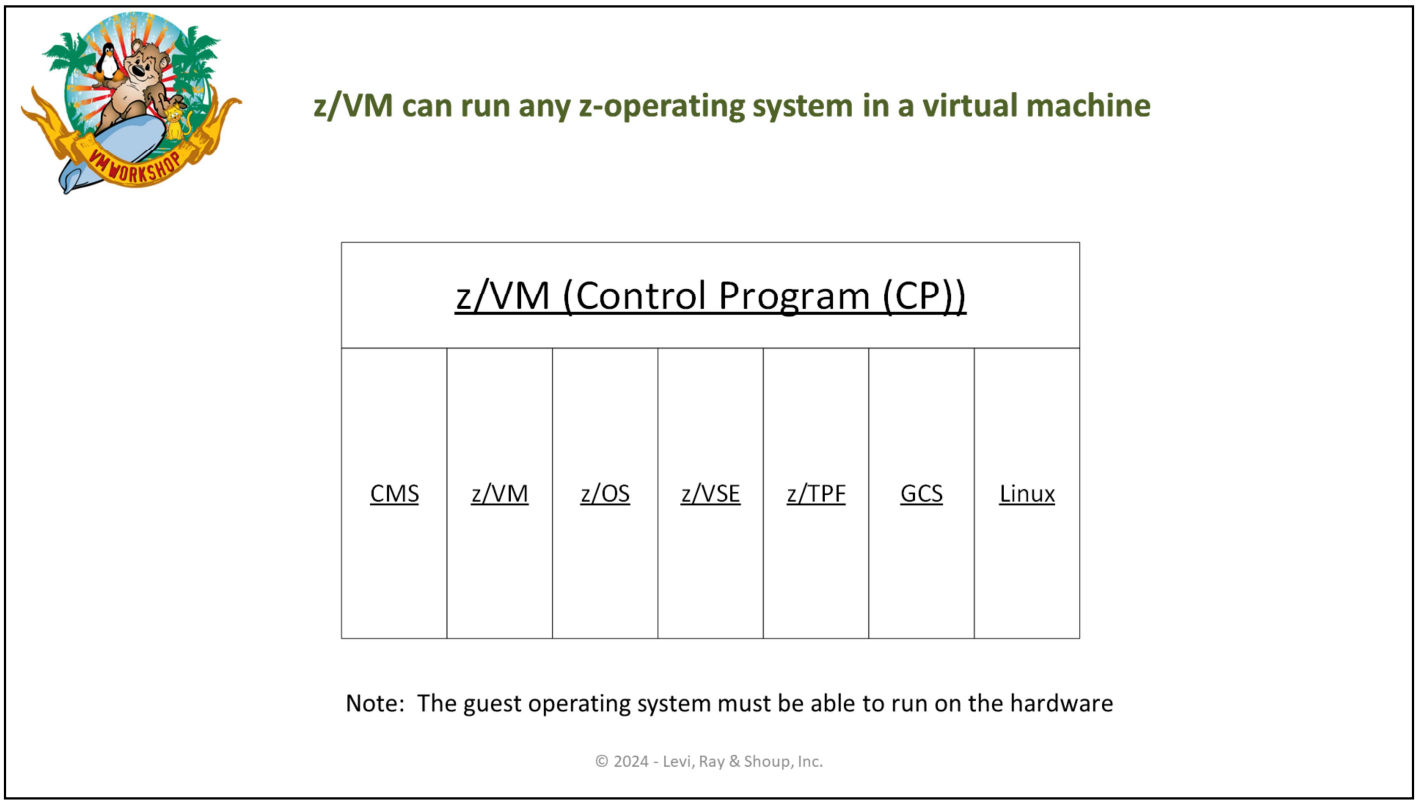
The System Configuration File will be referenced in a later chart when IPL of z/VM is discussed

VM Directory

- Defines every virtual machine that can be activated (logged on)
  - Identifies initial resource allocations
  - Authorizes certain CP commands and informational displays
- Created by the DIRECTXA command from source code
- Doesn't care if you overlap disk storage areas – use DIRMAP/DISKMAP EXECs to map out disk allocations from Directory Source file or use Directory Management tool
- No disassembler!
  - Don't lose your source



This example shows a virtual machine setup to run Linux using shared read-only minidisks. These minidisks (contained in the profile) contain executables (such as /boot and /usr) that can be shared between Linux instances and managed centrally.



Note that you can run a complete copy of z/VM in a virtual machine. This is good for testing new releases of z/VM or CP modifications prior to putting them into production.

z/VM Components

- Control Program (CP)
- Conversational Monitoring System (CMS)
- Group Control System (GCS)
- Dump Viewing Facility (DVF)
- REXX Procedure Language (REXX)
- VM Serviceability Enhancements – Staged/Enhanced (VMSES/E)
- APPC VM/VTAM Services (AVS)
- Transparent Services Access Facility (TSAF)
- Language Environment (LE)

We've talked about CP. CMS and GCS are operating systems that have a symbiotic relationship with CP and cannot run on their own. The other components are tools/middleware/libraries.

z/VM Components (cont'd)

- TCP/IP
- Hardware Configuration Definition / Manager (HCD/HCM)
- Systems Managed Storage (DFSMS/VM)
  - Not pre-installed, must be ordered as optional product with z/VM order
- Remote Spooling & Communications Subsystem (RSCS)*
- Directory Maintenance (DIRMaint)*
- Resource Access Control Facility (RACF)*
- VM Performance Toolkit (PerfTk)*

- * - Chargeable feature

DFSMS/VM is not the same as DFSMS for z/OS. IODF output from z/VM's HCD/HCM is not compatible for use by z/OS guests or z/OS in other LPARs. Note that you do not need to use HCD/HCM…if your real hardware IOCDS is being managed from another LPAR, you can allow z/VM to be fully dynamic in its I/O recognition by using the defaults coded in the SYSTEM CONFIG file.

z/VM Optional Program Products

- IBM Tape Manager for z/VM
  - Compatible with DFSMSrmm (z/OS) catalog
- IBM Backup and Restore Manager for z/VM
- IBM Operations Manager for z/VM
- High-Level Assembler
- IBM Cloud Infrastructure Center
- zSecure Manager for RACF z/VM
- others

Additional Optional Program Products may be available; see http://www.vm.ibm.com

Conversational Monitoring System (CMS)

- Interactive Operating System for z/VM
- Originally single-tasking, now multi-tasking
- Often used to setup environment for guest operating systems (like Linux)
  - Define/establish communication paths
  - Change virtual machine settings
  - Define/configure additional virtual resources
- Has a “console” mode and a “fullscreen” mode

This set of charts discusses CMS and its filesystems

CMS Filesystems

- Minidisk (original filesystem for CMS)
  - Explicitly defined portion of a physical disk (minidisk)
  - Cannot span physical disks
  - Access control at disk level
    - Based on Directory Entry
    - By knowledge of access password (read/write/multi-user)
  - Record-Oriented

A minidisk formatted for CMS usage could be considered analogous to a PDS, but you don't need to compress it

CMS Filesystems

- Shared File System (SFS)
  - Filepools under control of Shared File System server
  - Can span physical disks
  - Can create tree-structure (subdirectories)
  - Access control at file level (disk level is an option, not default)
  - Record-Oriented

SFS provides better access control for shared files and more efficient use of disk space, since filepools are shared between authorized users. Users are assigned size limits (quotas in Linux terms) which can be changed on-the-fly by the filepool administrator

CMS Filesystems

- Byte File System (BFS)
  - Similar to SFS (uses SFS servers for management of space)
  - Tree-structured directory
  - Similar to Unix-oriented file systems
  - Byte-oriented

BFS is used by the z/VM Shell and Utilities, which provides some level of POSIX compliance for z/VM. The z/VM LDAP Server uses BFS

CMS File Access

- CMS has 26 mount points (A-Z)
- Filesystems are mounted via ACCESS command
- Named files have an 8-character filename and 8-character filetype
- Similar to *filename.ext* in Windows
- Filemode refers to the mount point + a number to denote the allowed access
- Looking for a Resource:
  - No "Path" statement (no STEPLIB DD or PATH variable)
  - Mounted filesystems are scanned in sequence A-Z for the named file (unless filemode is specified)
  - Files are referenced by Filename + Filetype + Filemode (e.g. PROFILE EXEC A1)

The first entity to be found during the search will be used; i.e. if you are looking for PROFILE EXEC * and this file exists on the B, D and M disks, the one on the B-disk will be used.

Some Reserved CMS Filetypes

- EXEC – REXX or EXEC2 (shell script)
- HELPxxxx – Help-Related File
- LISTING – Output (Print) File
- MACLIB – Macro Library
- MODULE – Executable Object
- NAMES – CMS Nicknames
- TEXT – Relocateable Object
- TXTLIB – Text Library
- XEDIT – Editor Control File(s)

Standard CMS Mount Points, Esoteric

- 191 Disk or root SFS directory (.) A
- 192 Disk D
- 190 Disk (System Disk) S
- 19E Disk (System Disk Extensions) Y/S
- 18x Tape Drive TAPx

  (e.g. 181 = TAP1, 182=TAP2, etc)

The Y/S indication means that the Y-disk is a read-only extension of the S-disk, and that the Y-disk will be searched right after the S-disk (instead of searching T, U, V, W and X disks before searching the Y-disk). The TAPx is the only use of an esoteric in CMS; there is no other concept of an esoteric is CMS.

Command/Shell Processing

- Older script languages: EXEC and EXEC2
- Current script language: REXX

- REXX Scripts start with /* *comments* */ in line 1 of file
- Little distinction between referencing shell scripts (written in REXX, filetype=EXEC) and executables (filetype=MODULE)
- If CMS doesn't recognize a command, it passes it to CP for execution
  - **Note**: This is not true for other operating systems in a virtual machine

TSO requires the first line of a REXX proc to say "/* REXX */". This is unique to TSO REXX and is not required by any other REXX implementation

Common CP/CMS Commands

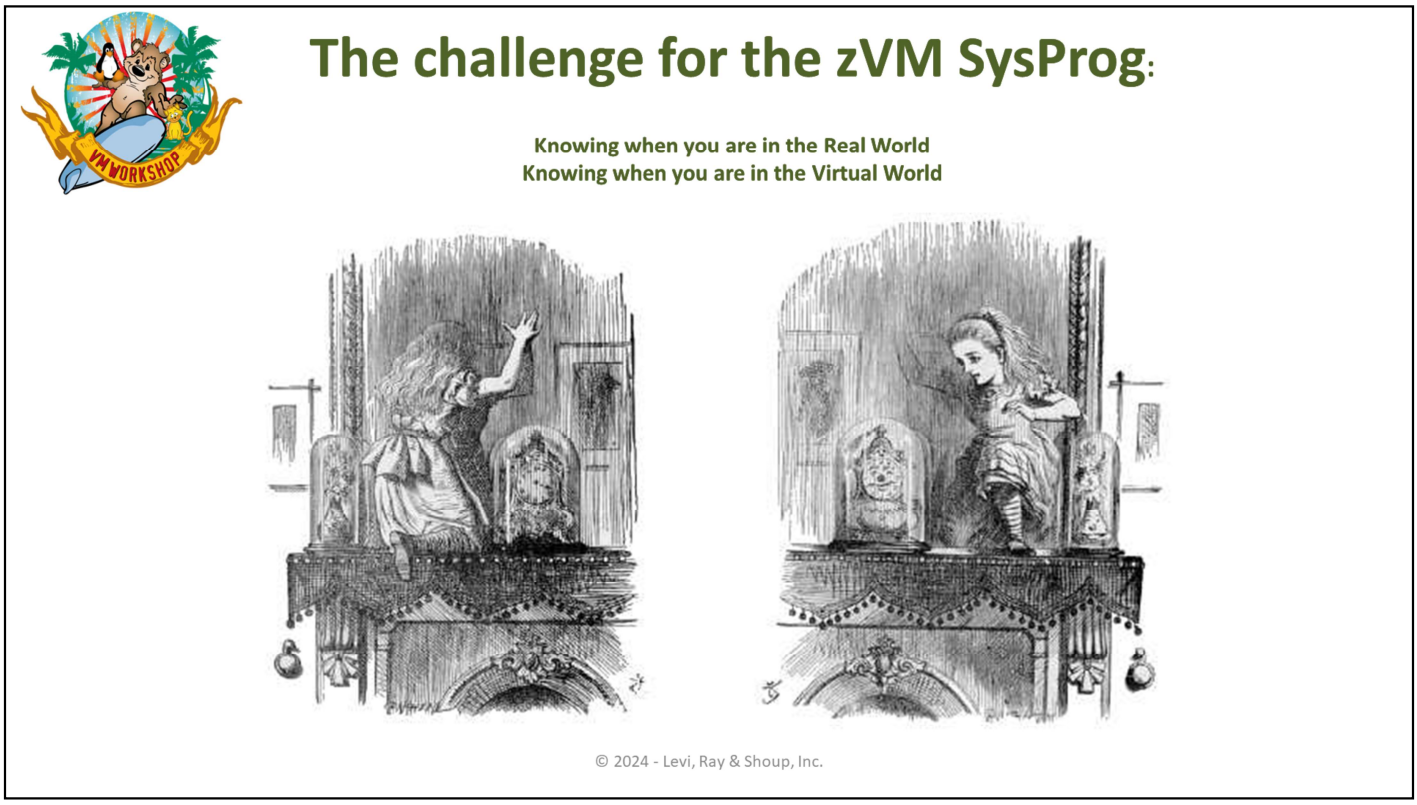
- HELP – when in doubt, type HELP
- Manage Real Resources for a Virtual Machine (CP)
  - LINK – connect a minidisk to your virtual machine
  - DEFINE – add a virtual device to your virtual machine
  - ATTACH – connect a real device to your virtual machine
  - DETACH – remove a device from your virtual machine
- Manage file availability to a CMS User (CMS)
  - ACCESS – add filesystem to search order
  - RELEASE – remove filesystem from search order
  - VMLINK – general purpose link/access/release/detach exec
- File Maintenance (CMS)
  - FILELIST – view files based on search criteria
  - RDRLIST – view files in virtual card reader
  - XEDIT – native editor of CMS
  - BROWSE – read-only file viewer

HELP contains messages and codes, command syntax and “how-to” info, all in one command. Note that some of the listed commands are CP commands, others are CMS commands

Common CP/CMS Commands (cont'd)

- Operating System Operation (CP)
  - LOGON – establish a virtual machine environment
  - IPL – start an operating system in a virtual machine
  - LOGOFF or LOGOUT – destroy the existing virtual machine environment
- Operating Environment Settings (CP/CMS)
  - QUERY – view environmental settings
  - SET – change environmental settings
  - **Note:** Ability to view/set values depends on virtual machine authorizations

Remember that if you type LOGOUT or LOGOFF when CP is looking for a command (CP READ in the lower right corner of the console), then the virtual machine goes away. Linux virtual machine console users must be aware of their console mode (RUNNING, VM READ, CP READ, etc) before entering a command or the unexpected may occur.



If you are a z/VM Systems Programmer, you must know which side of the "looking glass" you are on…the real side or the virtual side. You must also know when you are requesting information on the real environment and when you are requesting information on the virtual environment.

Guest Operating System Considerations under z/VM

- VM Paging Planning
  - See z/VM Planning and Administration Manual for your release of z/VM
  - Depending on workload, can range from 1.2:1 to 2:1 Virtual/Real (higher for CMS users)
- Disk Space/Memory Planning
  - Exploit z/VM benefits to reduce real I/O
  - Linux multiple swap disks (one on virtual disk in storage, overflow to physical swap disks)
  - Minidisk caching for read-mostly disks
  - Tune your environment
  - Don't throw processors or memory at a virtual machine without analysis first
  - Your system may run faster with fewer virtual processors and less memory to manage
  - Turn on compiler optimization including use of the latest hardware instructions
  - Don't follow rules-of-thumb for other hardware and virtualization platforms
- HiperDispatch
  - Vertical Polarization may cause contention for dispatching on too-few logical processors
  - Horizontal Polarization may cause cache thrashing, disables multithreading

z/VM Paging changed a lot starting with z/VM 6.3. Use of Expanded Storage was normal prior to z/VM 6.3. Virtual-to-real planning is very dependent on the type of application you are running….database servers need to have all their 'defined' (to the database) memory available, while application servers may be ok to allow some overcommitment of memory.

Where to Get More Information

- VM Workshop sessions this week
- Speak with education providers here this week
- z/VM Website (https://www.vm.ibm.com)
- z/VM Education Roadmap (https://www.vm.ibm.com/education/)
- z/VM Adobe Indexed PDF Collection Kit (http://www.vm.ibm.com/library/)
- IBM RedBooks (http://www.redbooks.ibm.com)
- LinuxVM (http://www.linuxvm.org) (older but still helpful)

Questions?

More Details

CP Topics - Shared Segments

- Commonly-used code can be shared among all users
  - Termed Discontiguous Shared Segment (DCSS)
  - Operating Systems can be loaded into a DCSS and IPL'd using a Named Saved System (NSS)
  - CMS is an example of an NSS
- Similar in concept to Shared LPA (z/OS)

CP Topics – DASD Usage

- Allocation Byte Map
  - Cylinder 0, Head 0
  - 1 byte/cylinder (through 3390-27, different mapping for larger disks)
  - Used by CP-Owned Devices
  - Indicates usage of disk space (PERM, DRCT, PAGE, SPOL, TDISK, PARM)
  - Starting with z/VM Version 5, full-pack PAGE and SPOOL packs bypass the Allocation Byte Map area
    - You'll see allocation (PAGE or SPOOL) starting in cylinder 0

If you are planning to use full volumes for z/VM page/spool space, you must understand the role of the allocation byte map or strange and apparently random bad things may happen to your running system

CP Topics – System Startup

- IPL (Initial Program Load)
  - Specified device address on HMC/SE
  - Stand-Alone Loader started if console address is entered in LOADPARM
  - Looks in PARM area for CP Nucleus (module) and System Configuration File
  - Console is required **only** during CP IPL sequence
    - Can use HMC "Operating System Messages" (device SYSC) or "Integrated 3270 Console" (device SYSG)
    - These devices are already defined in the IBM-delivered z/VM base

The Allocation Byte Map is read by CP for all CP-OWNED volumes as listed in the System Configuration file. CP Dump Space is allocated in Spool based on the amount of real storage that CP sees at IPL time

CP Topics –System Startup (cont'd)

- IPL (cont'd)
  - Following CP IPL, five virtual machines are started:
    - EREP – gathers hardware/software error records
    - DISKACNT – gathers disk accounting records
    - OPERSYMP – gathers symptom records
    - OPERATOR – system interface or automation
    - AUTOLOG1 – autostarts other virtual machines
  - You can disable any or all of these from starting
    - z/VM does not require an "Operator" or a separate physical console device

You must plan for log management of EREP and DISKACNT space; most shops use the VMUTIL virtual machine for this via the WAKEUP service (similar to CRON)

CP Topics – Other Items

- Dynamic I/O
  - Can add any real device on-the-fly
  - Changes in I/O subsystem that include z/VM LPAR are communicated to CP
- Unlike z/OS, No I/O Definition File (IODF) needed
  - CP-managed disks (with minidisks) have no VTOC
- Command Classes
  - A-G Pre-defined (A = Highest Level, G = General User)
  - Can define your own classes
    - Consider putting FORCE and XAUTOLOG in their own class for Help Desk personnel
- Performance Tuning
  - SET SHARE (Relative/Absolute, Soft Cap/Hard Cap)
  - SET SRM (System Resource Measurement values)
  - Vertical vs. Horizontal Polarization

IODF is an option starting with z/VM 4.4; it is **not** required